A blackboard architecture for a hybrid CBR system for scientific software

Miltos Petridis, Brian Knight

School of Computing and Mathematical Sciences
University of Greenwich
London SE10 9LS, UK
m.petridis@gre.ac.uk, b.knight@gre.ac.uk

Abstract

This paper describes the use of a blackboard architecture for building a hybrid case based reasoning (CBR) system. The *Smartfire* fire field modelling package has been built using this architecture and includes a CBR component. It allows the integration into the system of qualitative spatial reasoning knowledge from domain experts. The system can be used for the automatic set-up of fire field models. This enables fire safety practitioners who are not expert in modelling techniques to use a fire modelling tool. The paper discusses the integrating power of the architecture, which is based on a common knowledge representation comprising a metric diagram and place vocabulary and mechanisms for adaptation and conflict resolution built on the Blackboard.

Introduction

Over the past years, advances in the computational power of modern computers combined with increased efficiency, accuracy and robustness of numerical modelling codes has transformed Computational Fluid Dynamics (CFD) software into a useful tool in a number of application areas in science and engineering. Nevertheless, the wide use of CFD software in many application fields has been hampered by the requirement for a high degree of expertise needed to set-up and run models. Early research by the authors pointed out the need for embedding expert knowledge into CFD software and demonstrated the feasibility of such an approach (Knight and Petridis 1992).

The expert knowledge associated with numerical modelling applications is typically multidisciplinary, combining knowledge of mathematics, numerical methods, physics and expertise in the particular application domain. Some of this knowledge can be difficult to elicit from the modelling experts and can be encoded in terms of past successful and unsuccessful cases. The use of a Case Based Reasoning Approach (CBR) (Kolodner 1992) within a CFD software package has been shown to be appropriate (Taylor 1997), but it needs to be combined with other AI techniques and paradigms to match the way that a modelling expert adapts past cases to the particular problem in hand and combines the adapted case with explicit knowledge of the underlying mathematical and physical processes to complete the task of setting up and running a model. Furthermore, an intelligent CFD system should allow the expert user to query and override decisions made by the system.

This paper reports the research associated with building such a system as part of the *Smartfire* project, sponsored by the UK Home office and the EPSRC. The main emphasis of this paper is on the software architecture of the intelligent component of *Smartfire* that allows the combination of a CBR system with other components into a hybrid CBR system that is used to decide on the setting up of fire field modelling problems.

The paper first presents the problem of fire modelling and the associated expertise required to set up a fire model. The architecture of the *Smartfire* system is presented. This is based on the blackboard architecture that allows for the various intelligent and numerical components to collaborate towards setting up and running a fire modelling exercise. Finally, a brief evaluation of the experience of building and evaluating the system is presented.

Fire Field Modelling

The research reported here deals with fire field models. Fire field modelling (Galea 1989) is a powerful and accurate technique, although it is expensive in terms of simulation time. Fire models are generally used to predict the growth and spread of a fire under given initial conditions. For example, to simulate the development of a fire in a room, the modeller needs to specify the room geometry and material composition, and the position and size of the fire before running the simulation. The outputs will show the spread of the fire. The theoretical basis of fire field modelling is Computational Fluid Dynamics (CFD). CFD software entails the iterative solution of equations balancing variables such as flow, mass, heat energy (enthalpy), pressure and other related elements. This solution is performed over a grid (or mesh) of volumetric cells representing the problem geometry. CFD may be applied to many problem domains from aerodynamics to liquid flow. There are a number of CFD packages on the market (FLOW3D 1991), (Spalding 1981). All CFD systems require a considerable amount of effort and expertise in the preparation of their inputs. This

Copyright © 2001 Miltos Petridis, Brian Knight

expertise is often not available to the potential users in the Fire Safety Engineering community, but it can be crucial to the usefulness of the simulation. Without expertise in modelling, the fire safety engineer may have great difficulty running the simulation and obtaining the best results in good time.

In order to incorporate some of the required expertise, the *Smartfire* project (Taylor et al 1997), (Ewer et al 1999), (SMARTFIRE 2001) has adopted an approach based on Case Based Reasoning (CBR) (Kolodner 1992). The expertise relates the input room geometry to the optimum initial setup for the CFD component of the package. The system deals with complex fire simulations while hiding much of the complexity associated with traditional fire field models. A graphical user interface allows specification of room geometries with varying fire situations. The system has been validated against other CFD simulations and data from experiments conducted by Steckler et al (1982). The example cases presented in this paper are based on these experiments.

Setting up the mesh

One of the most important aspects of setting up a fire model relates to imposing a mesh over the modeled area. The process of setting up a fire model involves expertise that is mainly related to the geometry of the model. Given a geometrical description of a room with a number of vents and a fire source, the expert needs to divide the domain into cells by imposing a mesh over the modelled domain. The mesh is used to solve the discretised partial differential equations that describe the complex physical phenomena taking place. A fire modelling expert can identify important regions in the domain that necessitate the use of many small cells and less important regions that a coarser mesh will be enough. A fine mesh cannot be imposed over the whole area, since this would often entail impractical run-times (typically in days or weeks). *Smartfire* is a fire modelling software package that aims to automate the set-up and running of fire models in an efficient and robust way. This necessitated the integration of expert knowledge into the system.

Early observations of experts showed that experts thought in terms of recalling and adapting past similar successful modelling cases. This was the main motivation that led to the idea of incorporating a CBR system into *Smartfire* that can encapsulate expert mesh generation knowledge. The knowledge acquisition process has taken a number of forms, from formal recorded interviews, through informal sessions answering questions, to cases. Knowledge acquisition began with a number of formal unstructured domain orientation interviews in order to elicit the major processes and domain factors. Following these, more detailed interviews were conducted to elicit specific meshing rules.

From these detailed interviews, general rules concerning grid generation emerged. Examples of these are the basic requirements of a good grid: fine cells near a wall, extended region for vents, etc. Attempts to elicit further general rules concerning meshing were generally met with the reply "depends on the position of the fire, the position of the vent, the size of the fire, the size of the vent, the aspect ratio of the room, etc.". This indicated that there were no further general rules, merely extremely specific ones. The decision was then made to continue the knowledge acquisition using *cases*. The expert was given a number of paper representations of room fire situations and asked to mesh given a particular cell budget. Figure 1 shows an example case. The figure is a scan of one of the actual cases used in knowledge acquisition. Given a cell budget of 25 cells in the X direction (anticipating the extended region) and a cell budget of 20 cells in the Z direction, the expert put the number of cells in each region along with the distribution of those cells. In the figure, this is represented as a number with an arrow. The number indicates how many cells in that region, the arrow indicates refinement of the cells in the direction of that arrow. Refinement is generally done using a power law, raising the number of the cell concerned to find its location in the geometry.

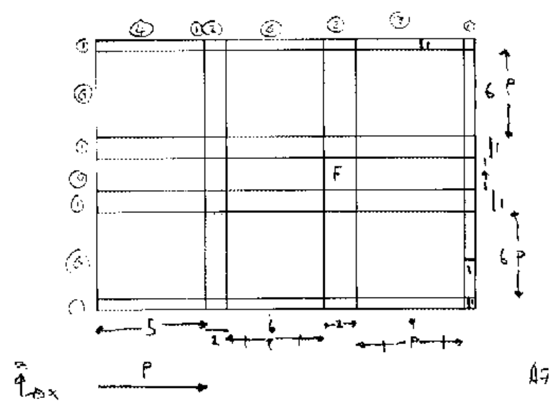

Fig. 1. : Example Case Received From Expert

The knowledge acquisition process consisted of giving the expert many sets of these cases, and after a number of these sets, patterns began to appear. The expert was referring back to previous cases in order to mesh a new case, adapting when necessary. This indicated that the expert was indeed thinking in a case based way.

The cases were then analysed in an attempt to induce cell allocation rules for groups of items. Some correlations were found and a test system implemented. However, this proved far too brittle to be effective as a reasoning system; when given a new case the system failed to produce an acceptable grid. The decision was then made to convert the rule-based system for mesh generation into a full case based reasoning system with all the advantages of matching the best cases, adaptation, learning, etc. This, coupled with the realisation that the expert was thinking in a case based way, made case based reasoning the most appropriate inference method.

Completing the set-up process

Retrieving a similar past case is just the start to the set-up process. Any retrieved past case must be adapted for the problem in hand using a number of heuristic rules. For example, the experts would use a number of basic guidelines for cell distribution. Distribution of cells for a particular region relies on a number of key parameters, common to all problems: there must be a fine cell next to a wall, and aspect ratios between cells must not be extreme. This means that the size of a cell compared to the size of its neighbour in a particular direction generally must be at most 3:1. A rule based subsystem proved applicable here. Figure 2 shows two examples of aspect ratios across cells. Both examples are constant aspect, i.e. the cells' size decreases at the same rate. From the figure, it can be seen that 1.5:1 is "smoother" than 2:1 (graduated more gently), and this is generally desirable. Rules checking these aspects attempt to keep the aspects as smooth as possible. For near wall areas and extended regions, a single power law is applied. Other rules check regions inside the geometry, and attempt to keep the aspects reasonable by applying a third order spline to the cells in the region, using the two outer cells of the region as the guide points.

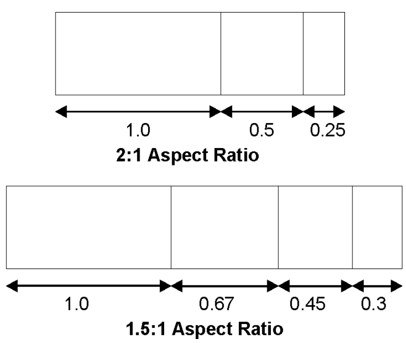

Fig. 2. Example of Cell Adjacency Aspect Ratios

Other guidelines are acquired that ensure that the grid is sensible. As an example, there must be at least two cells in the fire whatever the cell budget, because the fire is a concentrated source of a great deal of energy and momentum. To put this in a single cell will cause great instability due to the massive discontinuity on all faces of the fire cell. If this is not the case, then a cell is added.

Finally, the model set-up is completed by setting-up a number of other parameters and settings for the numerical solution. These can model:

- a physical aspect of the problem (e.g. the fact that there is no fluid flow through a wall or heat flow through an insulation)
- a solution process aspect, such as the time step for a transient problem, or the total simulated time
- a numerical aspect of the problem, such as the relaxation parameters error tolerance and number of iterations for convergence (Ewer et al 1999)

The decision on these settings relies on heuristics that relate to qualitative characteristics of the problem and to the mesh characteristics. For example, a qualitative description of the grid as "coarse" should trigger a "small" time step. A number of such dependencies must be satisfied before the start of the modelling run to achieve a successful outcome. The various intelligent components interact within *Smartfire* to achieve this. The enabling factor is the use of a flexible software architecture. This is presented in the following section.

The Blackboard Architecture

The successful collaboration of the various components in *Smartfire* relies on its flexible software architecture. From the beginning of this research project, the main requirements of such an architecture were defined as:

- The architecture should provide a unifying view to the problem through the provision of a common vocabulary. This is very important as the various system components such as the CBR system, the heuristic intelligent component, the numerical system and the user interface should be able to use common references to the various aspects of the system.
- There should be components for transforming between the various individual component views and the common vocabulary view.
- The architecture should have a mechanism for sequencing the various expert tasks required for setting-up a system (e.g. making sure that a mesh is chosen before deciding on its adaptation or a solution process).
- The architecture should have mechanisms for combining solutions and for reconciling possible conflicts between the various components.

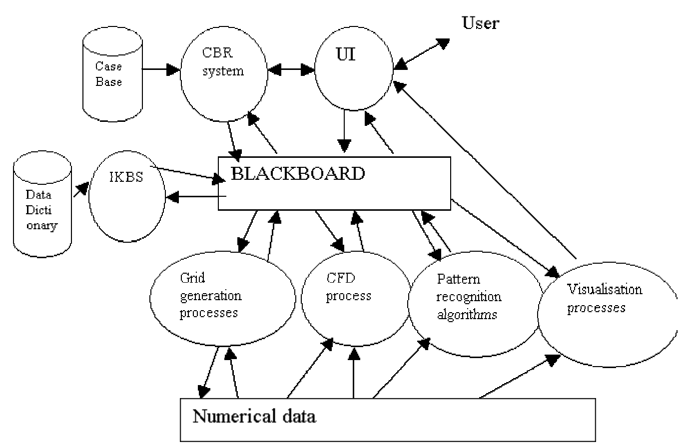

Fig. 3. The *Smartfire* software Architecture

It was decided to base the architecture on the blackboard architecture (Engelmore and Morgan 1986). Earlier research had shown the suitability of this architecture for CFD software (Petridis and Knight 1996).

The underlying paradigm for the blackboard (BB) as used in *Smartfire* is that of a shared working area, where the various intelligent components (CBR and IKBS in figure 3) in synergy with the numerical software and the user through the User interface come together to contribute to the model set-up. The Blackboard has a controller mechanism that has two main functions:

1. To sequence the contributions of the various components through the set-up process. This is achieved through a set of predefined states and required transitions of the set-up process.
2. To reconcile any "disagreement" between the various components. This is achieved through a hierarchy defined for each task. Invariably, the user through the user interface is at the top of this hierarchy as they can override any system decision as long as the user input is valid (physically possible)

The blackboard architecture uses a place vocabulary as the common representation of any knowledge associated with the system. This provides the common view of the state of the set-up process for the problem in hand. This is discussed in the next segment. Figure 4 shows a schematic of the internal components and functions within the blackboard.

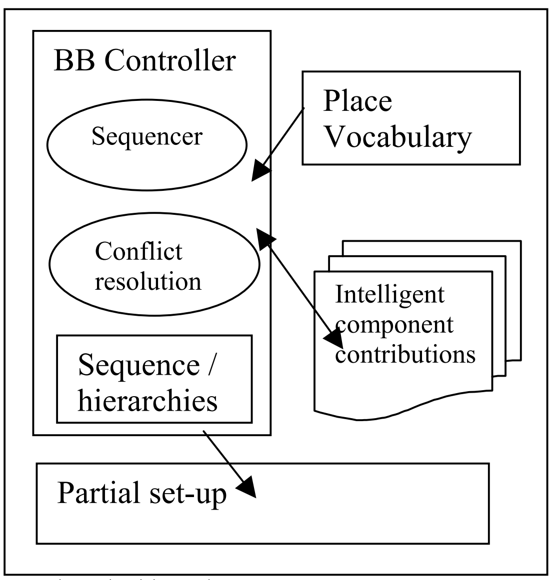

Fig. 4. The Blackboard

Knowledge Representation on the Blackboard

During the early elicitation stages of knowledge acquisition, the question of knowledge representation was investigated. An initial scheme was arrived at, based on these domain factors, for example fire, vent, room, near wall, etc. It was found that "belongs-to" relationships could be easily represented with standard entity-relationship models, for example: a room has many walls, each of which can have many vents. However, further knowledge acquisition showed that key parameters to the meshing process were relationships such as the relative location of fires and vents, and their relative sizes. It was found that this method of knowledge representation was not well suited in describing relationships such as these, because there was no view of the room and its contents as a whole.

Subsequent knowledge elicitation, acquiring knowledge using the problem sheets (such as that shown in Figure 1) into how the expert meshed certain problems, presented a view of the room and its contents. These problem sheets were developed into a representation scheme that has been dubbed the *block model*. The block model representation has two views: the *metric diagram* and the *place vocabulary*. These two views derive from the work of Forbus regarding spatial reasoning (Forbus, Nielsen and Faltings 1991). Forbus asserts that two views of spatial problems are necessary for representation. For this research, the metric diagram is a semi-quantitative representation, and the place vocabulary is a higher level, purely qualitative representation.

Metric Diagram

From figure 1, it can be seen that the cuts across the geometry governed by the physical objects such as the fire and the vent, etc., form a matrix of *blocks*. In the block model's metric diagram, each block describes a region of space that is qualitatively different from its neighbours in at least one way. Since the final grid is orthogonal, it was obvious that the representation of its "input", the geometry, should be orthogonal also. The metric diagram is a coarse qualitative representation of the final grid, describing two important factors: the constraints put on the grid by the physical objects, and expertise associated with certain regions in the geometry.

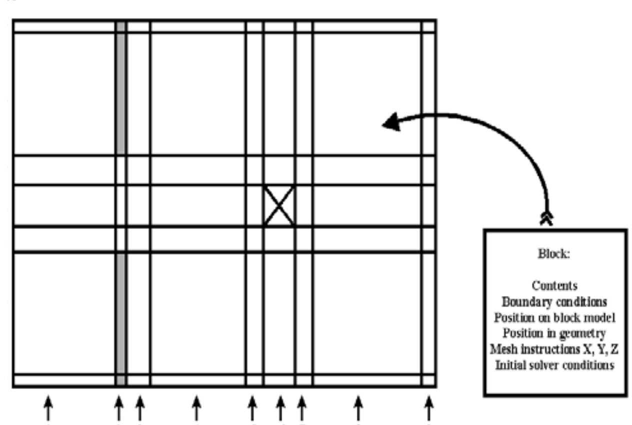

Fig. 5. Block Model Metric Diagram for Case A74

Figure 5 shows an example metric diagram in plan view for the A74 case. The block model is created by pattern recogniser functions. The geometry is read in each direction XYZ, and the pattern recognisers generate *critical points*. A critical point is a qualitative difference in the geometry in one particular direction. There are two types of critical point: *real*, for example the edge of a fire or vent or a wall, and *expert*, which are changes added by reference to expert knowledge. An important component of the block model is a *slice*. The concept of the metric diagram and slices emerged from the need to qualitatively represent the room's geometry in a way that would be comprehensible to both the expert system (for inference) and human experts (for knowledge acquisition). Indeed, the bulk of knowledge acquisition in this domain has used this representation.

Place Vocabulary

The place vocabulary is a high level qualitative representation of the information in the metric diagram. The primary use of the place vocabulary is for case retrieval. The case library and the current problem are both represented in this form. Attributes in the problem are compared against cases in the library to find the closest match for the current problem. Figure 6 shows a relational schema of the block model's place vocabulary view. The central object is the *room block model*. This is connected via a one-to-many relationship to a qualitative description of a particular combination of a single vent and a single fire. Knowledge acquisition has shown that important relationships are between vents and fires. This is because a fire is the key element in the room, and the vent directs flow.

A slice in each direction is also represented. These are the items that eventually receive meshing information from case retrieval.

Attributes of the vent fire relation entity describe the geometry of the room in qualitative, symbolic terms. For each attribute, there is a dataset containing a number of *discrete bands* (Petridis and Knight 1996). Such a dataset transforms continuous numerical data into qualitative discrete sets, with mnemonic terms attached. For example, consider the attribute *fire location*. This has the bands:

fire_on_wall
fire_close_to_wall
fire_away_from_wall
fire_centre

Each band has membership criteria, in this case the closeness of a fire to a wall in that particular direction. Each band in the dataset has a *score*, and this is used to compare the current problem with the stored case. Some datasets have more bands than others, for example *fire vent in line* has only two bands since it is a yes/no relationship.

The place vocabulary uses a concept named *vent normal*. This is key to the description since it represents the direction that a particular vent is open towards. Each vent has a normal direction. For example, in figure 3 the vent is facing X, i.e. it is looking in the X direction. The vent normal direction has a counterpart, *orthogonal*. In figure 5, the orthogonal direction is Z. This is the horizontal direction perpendicular to the vent facing direction. The case library is organised using the vent normal concept. In the place vocabulary, the direction normal to the vent is represented as VN. The orthogonal direction is VNO, and the vertical direction is VERT.

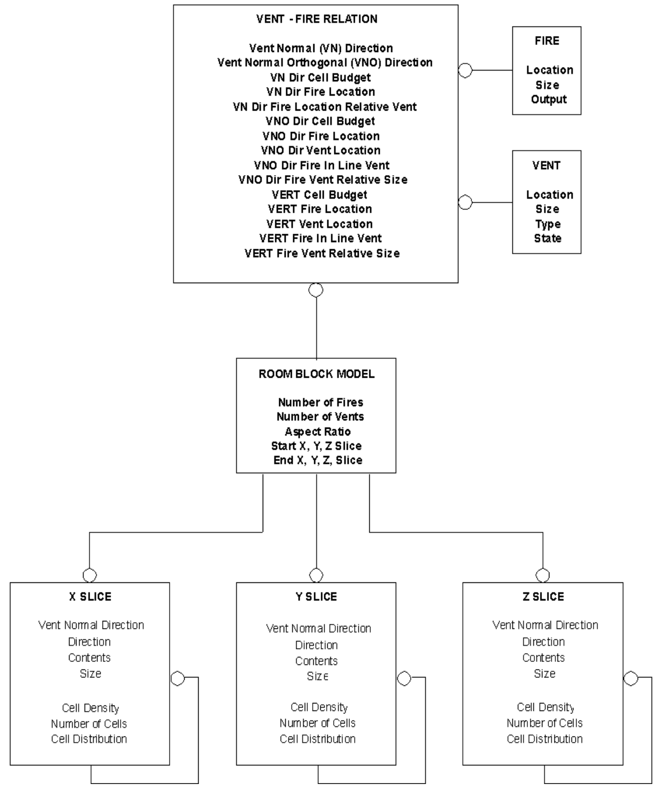

Fig. 6. Place Vocabulary

Other attributes in the direction description entities are further descriptive facts regarding the geometry. *fire vent relative size* indicates how much larger the fire is than the vent or vice versa. *vent location* indicates where a vent is in relation to the walls perpendicular to its wall, i.e. in its orthogonal direction. *cell budget* is the number of cells to be used in that particular direction. For the case library, the cell budget is the number of cells used in the validated case, and for new problems, a minimum cell budget is inferred using expertise. The user can then add more if they wish. *fire vent in line* determines if a fire is visible through a vent and will therefore cause direct flow through that vent. *fire location relative vent* is the relative locations of the fire and the vent. The fire can be on, close to, or away from the vent wall; centre; or on, close to, or away from the wall opposite the vent. All these attributes are banded using datasets described above.

There is a one-to-many relationship between the room block model and each set of slice descriptions. This is determined by how many slices there are in a particular direction of the block model. Each slice is described by its contents, which can be *extended region*, *fire*, *near fire*, *vent*, *near wall*, *wall*, *space* or a combination of these. Each slice can be between other slices, and this is represented by the one to many link from the slice type to itself.

CBR retrieval and adaptation process

The place vocabulary is the storage format used for the case library. The current problem is cast into the same form and the closest case in the library is retrieved to match it, using the attributes in the vent fire relation objects as indices. At the point of matching, many, but not all, attributes are filled. The contents of the retrieved case fill in the remaining attributes. Following this, a further process of adaptation makes sure that a number of additional rules are satisfied (such as the aspect ratios between neighbouring cells are not dramatically different). This is done in an iterative way on the Blackboard, until all associated heuristics are satisfied or a possible conflict is resolved on the Blackboard or by the user.

Finally, the intelligent system decides on additional settings on the Blackboard based on the characteristics of the retrieved (and adapted) mesh, the problem definition and any constraints imposed by the user (Knight, Petridis and Galea 2000). For example, a suitable time step will be proposed for a transient model based on criteria such as the cell budget of the constructed mesh, the required accuracy and the time constraints imposed by the user.

Evaluation of the system

The hybrid CBR system has been integrated successfully in the *Smartfire* software package. The system has been tested by non-experts on a family of realistic single-room fire modelling problems. The work funded by a combination of EPSRC and various industrial and government groups such as EU under FWkV, Rockwool, UK Home Office, LPC, etc has lead to the development of the *Smartfire* fire field model. This is one of the most innovative CFD based fire field models available, currently with users in 9 countries.

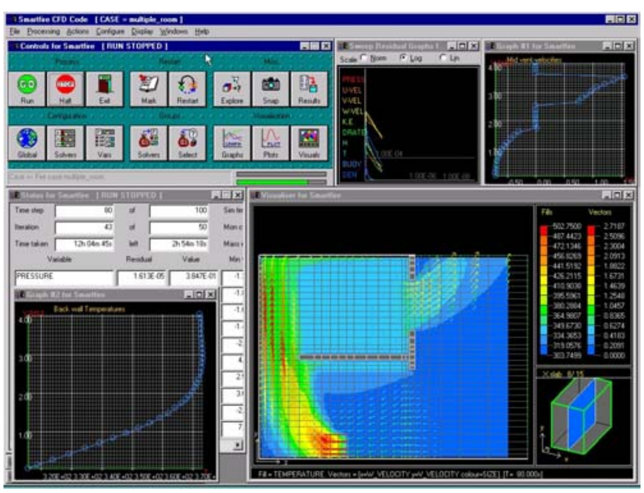

Fig. 7. Running a fire field modelling problem in *Smartfire*

Figure 7 shows an example of a fire modelling run session in *Smartfire.* The quality of the CBR generated meshes has been assessed using a set of 10 test problems. The solutions provided from these tests were compared to expert generated solutions and experimental data showed numerical agreement within acceptable tolerance (Taylor 1997). The efficiency of the modelling process in terms of additional time required to run a model (when compared to an optimised expert generated mesh) is easily offset by the time it takes an expert to generate a mesh manually compared to automatic generation. The *Smartfire* system is now used successful commercially and as an educational tool (SMARTFIRE 2001)

Conclusions

This paper has described the lessons learned from the design and implementation of a hybrid CBR system that generates input CFD set-ups for fire simulation situations.

A blackboard architecture has been used to provide the integration between the various parts of this system. This has proven to be a very flexible and extendible architecture that allows for the building of scaleable hybrid CBR systems. The key element of this architecture is the use of a common knowledge representation schema that allows for higher level communication and collaboration between the intelligent components, the numerical components and the user. The knowledge representation schema consists of two levels. These correspond to a low level, semi-quantitative representation (*metric diagram*), and a higher level qualitative representation describing relationships between objects in the geometry (*place vocabulary*). The metric diagram is meshed using the place vocabulary representation to retrieve library cases that determine cell density. The proposed solution is evaluated, and stored as a new case if different enough to the problem. Additionally, the blackboard architecture contains a controller module that allows for sequencing of activities and conflict resolution on the blackboard between the various components that contribute to the selection of a suitable set-up for a given problem, the CBR system being only one of these. A working system has been implemented demonstrating the ideas in this paper. The system produces reasonable meshes given a range of input geometries. The hybrid CBR component is currently being used successfully in both commercial and educational contexts as part of the *Smartfire* software package.

This research points at the suitability of the blackboard architecture for providing a standard integrating mechanism for hybrid CBR systems that integrate different computing paradigms into "soft" CBR systems. Experience in this research has shown that the blackboard can provide the abstraction needed to allow for the combination of knowledge and expertise from various paradigms. Furthermore, it provides a common repository and reasoning mechanism that allows the reconciliation of advice coming from the various intelligent components found in a hybrid system.

Further work is under way that aims to address the application of the ideas discussed in this paper to more complex room geometries containing multiple rooms. Also, there are plans to extend the use of the blackboard for the dynamic real-time control of the solution process.

References

FLOW3D (1991) *Release 2.3.3 Reference Guide*, CFD Dept AEA Harwell UK, Feb. 1991.

Engelmore, R., and Morgan, A. eds. (1986). *Blackboard Systems.* Reading, Mass.: Addison-Wesley.

Ewer,J, Galea, ER, Patel, MK, Taylor, S, Knight, B, Petridis, M, (1999) *SMARTFIRE: An Intelligent CFD Based Fire Model* Journal of Fire Protection Enginering, Vol 10 (1), pp. 13-27

Forbus, K., Neilsen, P., Faltings, B., (1991) *Qualitative Spatial Reasoning: the CLOCK Project*, Artificial Intelligence, Vol. 51, pp 417-471.

Galea E.R., (1989) *On the field modelling approach to the simulation of enclosure fires*, Journal of Fire Protection Engineering, Vol. 1, no. 1, pp. 11-22.

Knight B, Petridis M (1992) *"FLOWES: an intelligent computational fluid dynamics system"* `Engineering Applications of Artificial Intelligence.' Vol 5, No. 1, pp. 51-58, Ed. M.G.Rodd, Pergamon.

Knight B, Petridis M, Galea E, (2000) *A Case Based Reasoning System For Automatic Setup Of Fire Models* in Proceedings of the 5th UKCBR Workshop, Ed. B. Lees, pp 11-22, BCS SIGES.

Kolodner, J.L.,(1992) *An Introduction to Case Based Reasoning*, Artificial Intelligence Review, Vol. 6, no. 1, pp. 3-34.

M. Petridis, B. Knight (1996) "The Integration of an Intelligent Knowledge Based System into Engineering Software using the Blackboard Structure" in the journal `Advances in Engineering Software', 25, pp. 141-147

SMARTFIRE (2001) WWW site:
http://www.gre.ac.uk/research/cms/fire

Spalding, D.B.(1981), *A General Purpose Computer Program for Multi-Dimensional One- and Two- Phase Flow*, Mathematics and Computers in Simulations, North Holland (IMACS), Vol. 23, p. 267.

Steckler, K.D., Quintiere, J.G., and Rinkinen, W.J., (1982) *Flow Induced by Fire in a Compartment*, NBSIR 82-2520, National Bureau of Standards, Washington, September 1982.

Taylor S J (1997) An Investigation into automation of Fire Field Modelling Techniques, PhD Thesis, University of Greenwich.

Taylor, S., Galea, E.R., Patel, M., Petridis, M., Knight, B., and Ewer, J.,(1997) *SMARTFIRE: An Integrated Computational Fluid Dynamics Code and Expert System for Fire Field Modelling*, Fifth International Symposium on Fire Safety Science, Melbourne, Australia, March 1997, pp. 1285-1296.